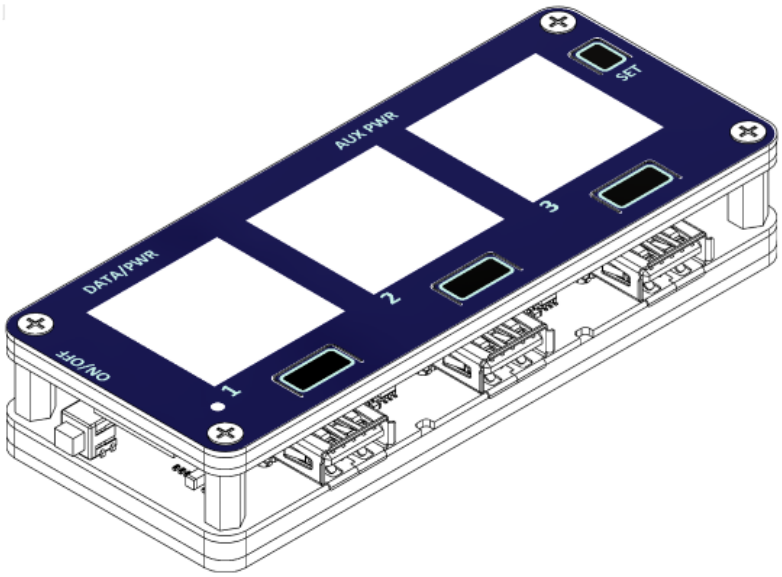
USB Insight Hub

SERIAL API v1.0



PN: USB-INSIGHT-HUB-A1

Doc Rev: v1.0A – September 2025

Aerio Solutions

Table of Contents

1. SERIAL API INTRODUCTION

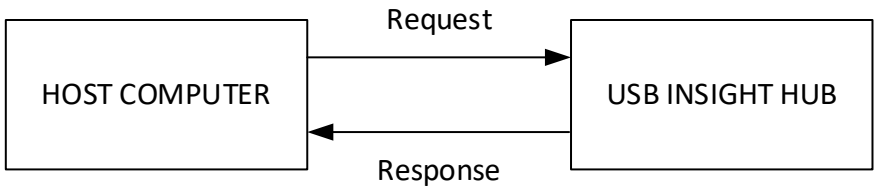
The serial API is the protocol used to send and receive information from the host to the ESP32 microcontroller in the USB Insight HUB (UIH). It operates and uses similar syntax as JSON-RPC with a simplified structure.

This mechanism is the one used by the Enumeration Extraction Agent to send the information to the displays in the UIH and for this reason, if the user needs to use the serial API, the Enumeration Extraction Agent must be disabled to free the virtual serial port.

2. OPERATION

2.1 COMMUNICATION MECHANISM

The host always makes a call to the UIH and receives a response. There are no notifications from the UIH to the host (unsolicited messages).



The communication channel is set to 115200 bps and DTR must be activated by the host computer to receive data from USB Insight Hub.

2.1 SET MESSAGES

Set – Used to change the value of one or more parameters.

```
{"action":"set","params": list of set params and values}
```

The *list of set params and values* is a json string with a structure of *parameters: value* pairs. Several parameters are part of a higher order parameter, for instance, each channel (CH1, CH2 and CH3) have several parameters associated to them.

Examples

First level parameter set request.

```
{"action":"set","params":{"ledState":"false"}}
```

```
Response:
```

```
{“status”:”ok”,”data”:{“valid”: “1 of 1”}}
```

Second level parameter set request:

```
{"action":"set","params":{"CH1":{"powerEn":"true"},"CH2":{"powerEn":"true"
}}}
```

Response:

```
{“status”:”ok”,”data”:{“valid”:“2 of 2”}}
```

2.2 GET MESSAGES

Get – Used to get the value of one or more parameters.

```
{"action":"get","params": list of get params}
```

The list of get params operates in a different way that the set messages: only first level parameters can be recalled (get) individually or with a get group parameter (all, conf, state). For CHx get parameters only group parameter works (all, CHx, CHx_all),

Examples:

Single parameter get request

```
{"action":"get","params":["hubMode"]}
```

Response:

```
{“status”:”ok”,”data”:{“hubMode”:“usb2”}}
```

Group parameters get request:

```
{"action":"get","params":["CH1"]}
```

Response:

```
{"status":"ok","data":{"CH1":{"voltage":"5019.9","current":"20.1","fwdAler
t":false,"backAlert":false,"shortAlert":false,"dataEn":false,"powerEn":tru
e}}}
```

Several parameters get request:

```
{"action":"get","params":["hubMode","ledState"]}
```

Response:

```
{"status":"ok","data":{"hubMode":"usb2","ledState":false}}
```

3. PARAMETERS

3.1 Global configuration set parameters (first level)

Parameter	Action	level	get group	values
startUpmode	get / set	root	all conf	"persistance", "on_at_start", "off_at_start", "sequence"
wifi_enabled	get / set	root	all conf	“true” “false”
hubMode	get / set	root	all conf	"usb2&3" "usb2" "usb3"
filterType	get / set	root	all conf	"moving_avg" "median"
refreshRate	get / set	root	all conf	"0.5s" "1.0s"
rotation	get / set	root	all conf	"0" "90" "180" "270"
brightness	get / set	root	all conf	“10” – “100” in %
ledState	get / set	root	all state	“true” “false”

3.2 System state get parameters (first level)

Parameter	Action	level	get group	values
startUpActive	get	root	all state	“true” “false”
pcConnected	get	root	all state	“true” “false”

Vbus	get	root	all state	Vbus voltage in mV
vext_cc	get	root	all state	Source power: “0” – Unknown “1” – Default 1A “2” – 1.5A “3” – 3.0A
vhost_cc	get	root	all state	Same as vext_cc
vext_stat	get	root	all state	“0” - CC1 nor CC2 has pull-up “1”- CC1 has the pull-up “2”- CC2 has the pull-up “3”- CC1 and CC2 are pulled up at the same time or CC1 or CC2 voltages are out of range.
vhost_stat	get	root	all state	Same as vext_stat
usb3_mux_out_en	get	root	all state	“true” “false”
usb3_mux_sel_pos	get	root	all state	“true” “false”
base_ver	get	root	all state	STM8 version (0-255)
esp32_ver	get	root	all state	ESP32 version (string)
mac	get	root	all state	ESP32 MAC (string)
pacRev	get	root	all state	Power meter revision (0-255)
cpu_freq	get	root	all state	
firstStart	get	root	all state	“true” “false” Changes true to false after first read.

menuIsActive	get	root	all state	“true” “false”

3.3 Channel set parameters (CHx level)

Parameter	Action	level	get group	values
powerEn	set	CHx	all CHx CHx_all	“true” “false”
dataEn	set	CHx	all CHx CHx_all	“true” “false”
startup_tmr	set	CHx	all CHx_all	“1” – “100” (0.1- 10.0 sec)
fwdLimit	set	CHx	all CHx_all	“100” – “2000” (in mA)
backLimit	set	CHx	all CHx_all	“1” – “200” (in mA)
numDev	set	CHx	all CHx_all	“0” – No devices “1” – One device “2” – Two devices “10” – Special Format. See Section 3.5
Dev1_name	set	CHx	all CHx_all	name string
Dev2_name	set	CHx	all CHx_all	name string
usbType	set	CHx	all CHx_all	“2” – D+/D- connection “3” – Super Speed connection

3.4 Channel get parameters (CHx level)

Parameter	Action	level	get group	values
voltage	-	CHx	all CHx CHx_all	“0.0” – “9000.0” in mV
current	-	CHx	all CHx CHx_all	“-2500.0” – “2500.0” in mA
fwdAlert	-	CHx	all CHx CHx_all	“true” “false”
backAlert	-	CHx	all CHx CHx_all	“true” “false”
shortAlert	-	CHx	all CHx CHx_all	“true” “false”
ilim	-	CHx	all CHx_all	Hardware short limit (automatically selected) “0” – 0.5A “1” – 1.0A “2” – 1.5A “3” – 2.0A
startup_cnt	-	CHx	all CHx_all	“1” – “100” (0.1- 10.0 sec)

* x = 1,2 ,3

3.5 Display enumeration text special format (numDev = 10)

In normal operation, when numDev = 1 or 2, the text in Dev1_name and Dev2_name are placed in one or two lines respectively.

Command	
{"action":"set","params": {"CH1": {"Dev1_name":"COM10","numDev":"1","usbType":"2"}}}	
{"action":"set","params": {"CH1": {"Dev1_name":"COM10","Dev2_name":"D:","numDev":"2" , "usbType":"2"}}}	

When numDev = 10, the Dev1_name has a json-formated string that gives more flexibility on the text displayed on screen. One (T1), two (T1 and T2) or three (T1, T2, T3) lines can be printed with a maximum of 14 characters per line.

Parameter	Action	level	get group	values
T1	set	Dev1_name	-	Text for line 1
T2	set	Dev1_name	-	Text for line 2
T3	set	Dev1_name	-	Text for line 3
txt	set	Tx	-	String for text 1 – 14 chars
align	set	Tx	-	“left” “right” Centered if this param is not provided
color	set	Tx	-	“YELLOW” “ORANGE” “BLACK” “RED” “DARKGREY” “CYAN” “BLUE” “GREEN” WHITE if this param is not provided

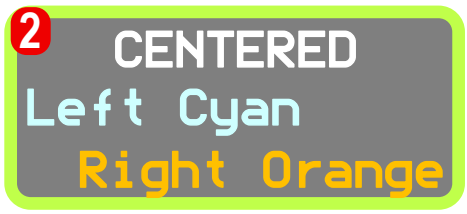
* x = 1,2 ,3

Examples

Command	
{"action":"set","params": {"CH2": {"Dev1_name":{"T1":{"txt":"CENTERED"}}, "numDev":"10", "usbType":"3"}}}}	
{"action":"set","params": {"CH2": {"Dev1_name":{"T1":{"txt":"CENTERED"}, "T2":{"txt":"Left Cyan","align":"left","color":"CYAN"}}, "numDev":"10", "usbType":"3"}}}}	

<table>
<tr>
<td>

```
{"action":"set","params":
{"CH2":
{"Dev1_name":{"T1":{"txt":"CENTERED"},
"T2":{"txt":"Left
Cyan","align":"left","color":"CYAN"},
"T3":{"txt":"Right
Orange","align":"right","color":"ORANGE"}},
"numDev":"10",
"usbType":"2"}}}}
```

</td>
<td>


</td>
</tr>
</table>